MEMBANDINGKAN ALGORITMA *DIJKSTRA* DAN ALGORITMA *FLOYD-WARSHALL* UNTUK MENENTUKAN LINTASAN TERPENDEK PADA PENDISTRIBUSIAN KORAN RIAU POS

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Sains
pada Jurusan Matematika

0leh :

M. JAMIL
10454025654

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2011

MEMBANDINGKAN ALGORITMA *DIJKSTRA* DAN ALGORITMA *FLOYD-WARSHALL* UNTUK MENENTUKAN LINTASAN TERPENDEK PADA PENDISTRIBUSIAN KORAN RIAU POS

M.JAMIL
10454025654

Tanggal Sidang : 30 Juni 2011
Periode Wisuda : November 2011

Jurusan Matematika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau
Jl. HR. Soebrantas No. 155 Pekanbaru

ABSTRAK

Permasalahan lintasan terpendek dalam suatu jaringan transportasi, merupakan suatu jaringan yang menghubungkan tempat asal ke tempat tujuan melalui rute-rute tertentu, dimana setiap rute-rute perjalanan tersebut diberikan bobot atau nilai. Skripsi ini akan membahas tentang perbadingan antara algoritma *Dijkstra* dan algoritma *Flyod-Warshall* untuk menentukan lintasan terpendek pada pendistribusian Koran Riau Pos. Berdasarkan hasil penelitian diperoleh bahwa, algoritma *Dijkstra* lebih efisien dibandingkan algoritma *Floyd-Warshall,* karena penelusuran simpul-simpul selalu mencari simpul yang berbobot minimum. Lintasan terpendek dengan algoritma *Dijkstra* dan algoritma *Floyd-Warshall* mendapatkan nilai yang sama yaitu 6500 Meter.

Kata kunci : *Algoritma Dijkstra, Aalgoritma Flyod-Warshall, Llintasan Terpendek.*

DAFTAR ISI

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Graf adalah himpunan simpul yang dihubungkan dengan suatu garis dimana garis tersebut menghubungkan dengan tepat ke 2 simpul sehingga simpul-simpul ini saling berhubungan. Berdasarkan kehidupan sehari-hari banyak sekali persoalan yang diimplementasikan dengan graf. Bidang-bidang yang menggunakan penerapan graf antara lain, *operation research*, aljabar, *computer science* dan kimia. Banyak sekali aplikasi yang mengunakan graf sebagai alat untuk merepresentasikan atau memodelkan persoalan sehingga persoalan itu dapat diselesaikan dengan baik.

Aplikasi-aplikasi tersebut misalnya menentukan lintasan terpendek *(the shortest path problem*), *traveling salesman problem, chinese postman problem*, graf *colouring,* dan masih banyak lagi. Kesempatan ini kami mencoba mengulas tentang persoalan menentukan lintasan terpendek (*the shortest path problem*). Menurut teori graf, persoalan lintasan terpendek adalah suatu persoalan untuk mencari lintasan antara dua buah simpul pada graf berbobot yang memiliki gabungan nilai jumlah bobot pada sisi graf yang dilalui dengan jumlah yang paling minimum. Persoalan lintasan terpendek ini banyak sekali dijumpai, Aplikasi yang paling sering ditemui adalah pada bidang transportasi, seperti pada pencarian rute terpendek untuk menempuh dua kota.

Berdasarkan kehidupan nyata sering ditemukan lintasan dari satu kota ke kota lainnya memiliki beberapa lintasan yang dapat dipilih. Misalnya, pada PT. Riau Pos akan mendistribusikan barang produksinya ke sejumlah *n* toko, dari toko yang satu ke toko yang lainnya masing-masing pada setiap toko satu unit. Seperti mendistribusikan barang produksi berupa koran dari toko satu ke toko lainya terdapat beberapa alternatif lintasan dan masing-masing lintasan mengandung resiko biaya dan waktu. Muncullah masalah menetukan pilihan lintasan, sehingga biaya dan waktu dapat ditekan dengan serendah-rendahnya.

Adapun teknik pemecahan diselesaikan dengan menggunakan metode yang ditemukan oleh *Dijkstra* (1959) dan disebut algoritma *Dijkstra*. Sedangkan metode lainnya adalah metode yang ditemukan oleh *Warshall* (1962) yang disebut algoritma *Flyod-Warshall*. Algoritma *Dijkstra* penentuan pilihan lintasan diperlihatkan dalam bentuk fungsi, sedangkan pada algoritma *Flyod-Warshall* dalam bentuk elemen matriks.

Berdasarkan latar belakang tersebut, penulis tertarik untuk mengambil judul tugas akhir yaitu ”MEMBANDINGKAN ALGORITMA *DIJKSTRA* DAN ALGORITMA *FLOYD-WARSHALL* UNTUK MENENTUKAN LINTASAN TERPENDEK PADA PENDISTRIBUSIAN KORAN RIAU POS”.

1.2 Rumusan Masalah

Permasalahan yang akan dibahas pada tugas akhir ini adalah bagaimana menentukan lintasan terpendek pada pendistribusian koran Riau Pos dengan membandingkan algoritma *Dijkstra* dan algoritma *Floyd-Warshall.*

1.3 Tujuan dan Manfaat Penelitian

1.3.1 Tujuan Penelitian

Penelitian ini bertujuan menentukan lintasan terpendek pada pendistribusian koran Riau Pos dengan membandingkan algoritma *Dijkstra* dan algoritma *Floyd Warshall.*

1.3.2 Manfaat Penelitian

Berdasarkan rumusan masalah dan tujuan penelitian yang telah dikemukakan, maka manfaat yang dapat diambil adalah sebagai berikut :

a. Penulis berharap dapat mengembangkan wawasan keilmuan dalam bidang matematika, khususnya tentang graf.
b. Penulis dapat memahami lebih mendalam tentang graf, sebagai bahan studi kasus bagi pembaca dan acuan bagi mahasiswa, terutama bagi yang ingin melakukan penelitian sejenis, juga menambah khasanah perpustakaan yang akan berguna bagi pembaca.

1.4 Sistematika Penulisan

Sistematika dalam penulisan tugas akhir ini mencakup lima bab, yaitu :

BAB I Pendahuluan

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan penelitian, dan sistematika penulisan.

BAB II Landasan Teori

Bab ini berisi informasi tentang teori-teori yang digunakan dalam penulisan ataupun metode atau algoritma yang digunakan dalam penulisan tugas akhir ini.

BAB III Metodologi Penelitian

Bab ini berisikan langkah-langkah dalam menyelesaikan lintasan terpendek pada pendistribusian koran Riau Pos dengan menggunakan algoritma *Dijkstra* dan algoritma *Floyd-Warshall.*

BAB IV Pembahasan dan Analisa

Bab ini berisikan penyelesaian perbandingan algoritma *Dijkstra* dan algoritma *Floyd-Warshall* dalam menentukan lintasan terpendek pada graf .

BAB V Penutup

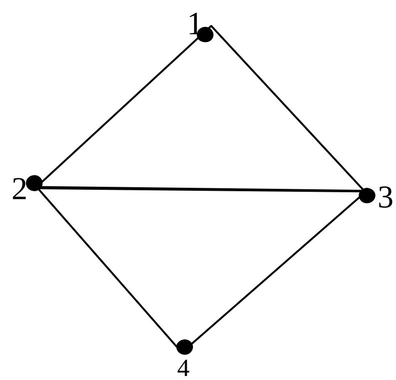
Bab ini berisikan kesimpulan dan saran.

BAB II
LANDASAN TEORI

Landasan teori yang penulis gunakan dalam penyusunan tugas akhir yang berjudul "**Membandingkan algoritma *Dijkstra* dan algoritma *Floyd-Warshall* untuk menentukan lintasan terpendek pada pendistribusian Koran Riau Pos"** adalah sebagai berikut:

2.1 Graf

Definisi 2.1 (Munir, Rinaldi : 2001) Graf G didefinisikan pasangan himpunan (V, E), dimana V adalah himpunan simpul-simpul (*vertex*) tidak kosong dan E adalah himpunan sisi (*edge*) yang menghubungkan sepasang simpul di V, ditulis $G = (V, E)$.



Gambar 2.1 Graf sederhana

dengan : $V = \{1,2,3,4\}$, $E = \{(1,2),(1,3),(2,3),(2,4),(3,4)\}$

2.2 Jenis-Jenis Graf

Definisi 2.2 (Munir, Rinaldi : 2001) Ada tidaknya gelang atau sisi ganda pada suatu graf, maka graf digolongkan menjadi dua jenis :

2.2.1 Graf Sederhana (*simple graph)*

Graf yang tidak mengandung gelang (*loop*) maupun sisi ganda dinamakan graf sederhana.

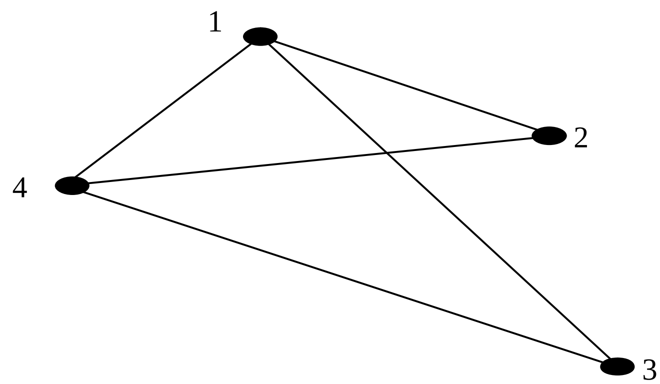
2.2.2 Graf Tak Sederhana (*unsimple-graph*)

Graf yang mengandung sisi-ganda atau gelang dinamakan graf tak-sederhana.

Berdasarkan dari jumlah simpul pada suatu graf, maka secara umum graf dapat digolongkan menjadi dua jenis :

2.2.3 Graf Berhingga (*limited graph*)

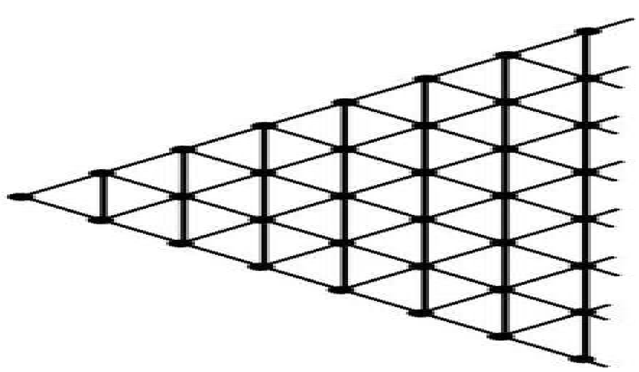
Graf berhingga adalah graf yang jumlah simpulnya n, berhingga.



Gambar 2.2 Graf hingga

2.2.4 Graf Tak-Hingga (*unlimited graph*)

Graf yang jumlah simpulnya n, tidak berhingga banyaknya disebut graf tak-hingga.

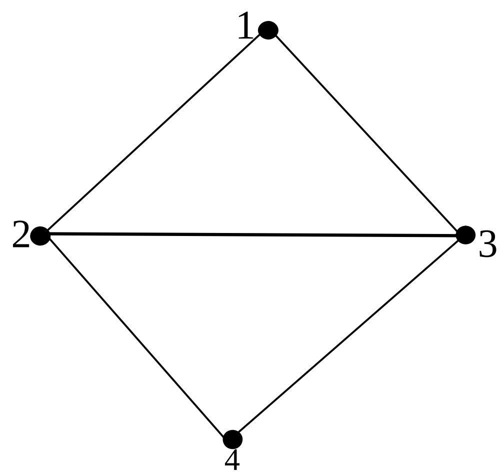


Gambar 2.3 Graf tak-hingga

Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas dua jenis :

2.2.5 Graf Tak-Berarah (*undirected graph*)

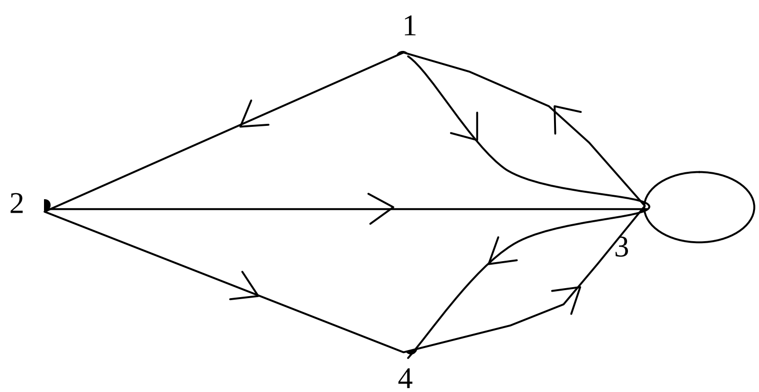
Graf yang sisinya tidak mempunyai oriantasi arah disebut graf tak-berarah.



Gambar 2.4 Graf tak-berarah

2.2.6 Graf Berarah (*directed graf atau digraph*)

Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah.

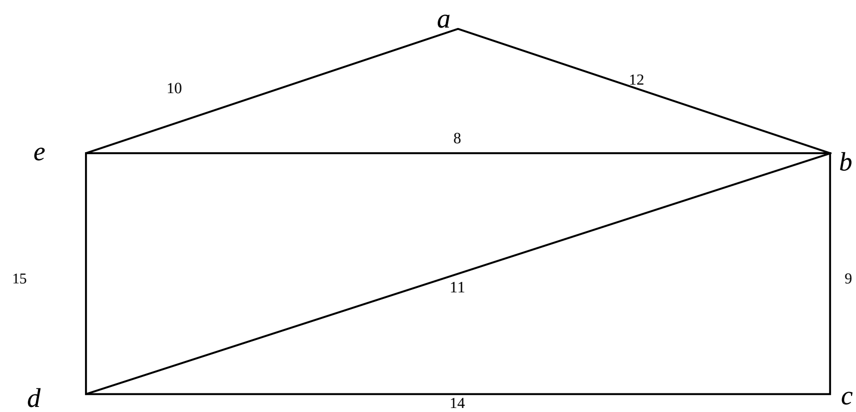


Gambar 2.5 Graf berarah

2.2.7 Graf berbobot/berlabel

Graf berlabel/ berbobot adalah graf yang setiap sisinya mempunyai nilai/bobot berupa bilangan, tetapi bukan bilangan negatif.

Contoh :



Gambar 2.6 Graf berbobot/berlabel

2.3 Beberapa Istilah yang Berkaitan dengan Graf

a. *Loop*

loop adalah suatu sisi yang menghubungkan suatu simpul dengan dirinya sendiri.

b. Lintasan

lintasan yang panjang n dari simpul awal v_0 ke simpul tujuan v_n di dalam graf G ialah barisan berselang-seling simpul-simpuldan sisi-sisi yang berbentuk $v_0, e_1, v_1, e_2, v_2, .., v_{n-1}, e_n, v_n$ sedemikian hingga $e_1 = (v_0, v_1), e_2 = (v_1, v_2), ..., e_n = (v_{n-1}, v_n)$ adalah sisi-sisi dari graf G.

c. Lintasan tertutup atau sirkuit

lintasan yang berawal dan berakhir pada simpul yang sama disebut lintasan tertutup.

2.4 Pengertian Lintasan Terpendek

Bobot yang berhubungan dengan suatu garis pada graf juga dapat diaplikasikan pada graf berarah. Prinsip dan arti bobot pada graf berarah sama dengan bobot pada graf tak berarah, yaitu menyatakan seberapa kuat hubungan antara dua titik yang arahnya ditunjukkan dengan arah garis. Salah satu aplikasi graf berarah berlabel yang sering dipakai adalah mencari lintasan terpendek di antara dua titik.

Sebagai contoh, terdapat banyak jalan yang menghubungkan kota Yogyakarta ke Jakarta. Pertanyaan yang sering muncul adalah, ”Jalur mana yang paling dekat?” Jika kota-kota di Jawa Tengah dan Jawa Barat dinyatakan sebagai titik-titik, jalan yang menghubungkan kota-kota tersebut dinyatakan sebagai garis yang menghubungkan titik-titik, dan jarak antara dua kota dinyatakan sebagai bobot garis, maka masalah mencari jalur yang paling dekat antara dua kota adalah mencari lintasan terpendek antara dua titik yang dinyatakan kota-kota yang bersangkutan. Apabila masalahnya adalah mencari jalur tercepat (jalur terpendek belum tentu tercepat), lintasan terpendek tetap dapat digunakan dengan cara mengganti bobot garis sehingga menyatakan waktu perjalanan antar kota-kota yang dinyatakan sebagai titik-titik di ujung garis. Cara yang sama dapat dipakai apabila dikehendaki untuk mencari jalur termurah.

Matriks hubung W yang digunakan untuk menyatakan graf berarah berlabel sama dengan matriks yang digunakan untuk menyatakan graf berlabel, yaitu elemen-elemennya menyatakan bobot garis. Akan tetapi, secara umum matriks hubung untuk menyatakan graf berarah berlabel tidaklah simetris karena bobot garis dari titik v_i ke $v_j\left(=W(i, j)\right)$ tidak sama dengan bobot garis dari titik v_j ke $v_i\left(=W(j,i)\right)$, bahwa mungkin hubungannya hanya searah, disamping itu, $W(i,i)=\infty$ untuk semua i.

2.5 Beberapa Algoritma (Metode) untuk Menentukan Lintasan Terpendek pada Graf

Beberapa algoritama (metode) untuk menghitung panjang lintasan terpendek pada sebuah graf yang akan diuraikan pada tulisan ini yaitu algoritma *Dijkstra* dan algorima *Floyd-Warshall*.

2.5.1 Algoritma *Dijkstra*

Algoritma *Dijkstra* yang ditemukan oleh E. W. *Dijkstra* adalah suatu metode untuk menghitung panjang lintasan terpendek merupakan algoritma yang lebih efisien dibandingkin algoritma *Warshall*.

Misalkan G adalak graf berarah berlabel dengan titik $V(G) = \{v_1, v_2, .., v_n\}$ dan lintasan terpendek yang dicari adalah dari v_1 ke v_n. Aloritma *Dijkstra* dimulai dari titik v_1. Sesuai dalam iterasinya, algoritma akan mencari satu titik yang jumlah bobotnya dari satu titik 1 terkecil. Titik-titik yang terpilih dipisahkan, dan titik-titik tersebut tidak diperhatikan lagi dalam iterasi berikunya.

Misalkan :

$V(G)$ = $\{v_1, v_2, \ldots\ldots\ldots.., v_n\}$

L = Himpunan titik-titik $\in V(G)$ yang sudah terpilih dalam alur lintasan terpendek.

$D(j)$ = Jumlah bobot lintasan terkecil dari v_1 ke v_j.

$w(i, j)$ = Bobot garis dari titik v_1 ke titik v_j.

$w^*(i, j)$ = Jumlah bobot lintasan terkecil dari v_1 ke v_j.

Secara formal, algoritma *Dijkstra* untuk mencari lintasan terpendek adalah sebagai berikut :

1. $L = \{\ \}$;

   $V = \{v_2, \ldots..v_n\}$;

   Untuk $i = 2, \ldots\ldots., n$, lakukan $D(i) = W(1, i)$.

2. Selama $v_n \notin L$ lakukan :

   a. Pilih titik $v_k \in V - L$ dengan $D(k)$ terkecil $L = L \cup \{v_k\}$

b. Untuk setiap $v_j \in V - L$ lakukan : Jika $D(j) > D(k) + W(k, j)$ maka ganti $D(j)$ dengan $D(k) + W(k, j)$

3. Untuk setiap $v_j \in V, w^*(1, j) = D(j)$.

Menurut algoritma di atas, lintasan dari titik v_1 ke v_n melalui titik titik dalam L secara berurutan, dan jumlah bobot lintasan terkcilnya adalah $D(n)$.

2.5.2 Algoritma *Floyd-Warshall*

Algoritma yang ditemukan oleh *Warshall* untuk mencari lintasan terpendek merupakan algoritma yang sedehana dan mudah implementasinya. Masukan algorima *Floyd-Warshall* adalah matriks hubung graf berarah berlabel, dan keluarnya adalah lintasan terpendek dari semua titik ke semua titik.

Langkah untuk mencari lintasan terpendek, algoritma *Floyd-Warshall* memulai iterasi dari titik awalnya, kemudian memperpanjang lintasan dengan mengevaluasi titik demi titik hingga mencapai titik tujuan dengan jumlah bobot yang seminimum mungkin. Misalkan W_0 adalah matriks hubung graf berarah berlabel mula-mula. W^* adalah matriks hubung minimal dengan $w_{ij}* =$ lintasan terpendek dari titik v_i ke v_j.

Algoritma *Floyd-Warshall* untuk mencari lintasan terpendek adalah sebagai berikut :

1. $W = W_0$
2. Untuk $k = 1$ hingga $n,$ lakukan :
   Untuk $i = 1$ hingga n lakukan ;
   Jika $W[i, j] > W[i,k] = W[k, j]$ maka tukar $W[i, j]$ dengan $W[i,k] + W[k, j]$
3. $W^* = W$.

Berdasarkan iterasinya, untuk mencari lintasan terpendek algoritma *Floyd-Warshall* membentuk n matriks sesuai dengan iterasi k. Hal ini menyebabkan waktu prosesnya lambat, terutama untuk n yang besar. Meskipun waktu prosesnya yang lambat, algoritma *Floyd-Warshall* sering dipergunakan untuk menghitung lintasan terpendek karena kesederhanaan algoritmanya. Tetapi proses implementasi algoritma *Floyd-Warshall* sangat sukar.

BAB III
METODOLOGI PENELITIAN

Metodologi yang penulis gunakan dalam penulisan tugas akhir ini adalah studi pustaka atau literatur, dengan membaca buku-buku dan sumber-sumber lain yang berhubungan dengan graf.

Langkah-langkah metodologi penelitian adalah sebagai berikut :

1. Memahami istilah-istilah dalam graf
2. Memahami definisi lintasan terpendek
3. Menentukan lintasan terpendek dengan algoritma :

   a. algoritma *Dijkstra* :

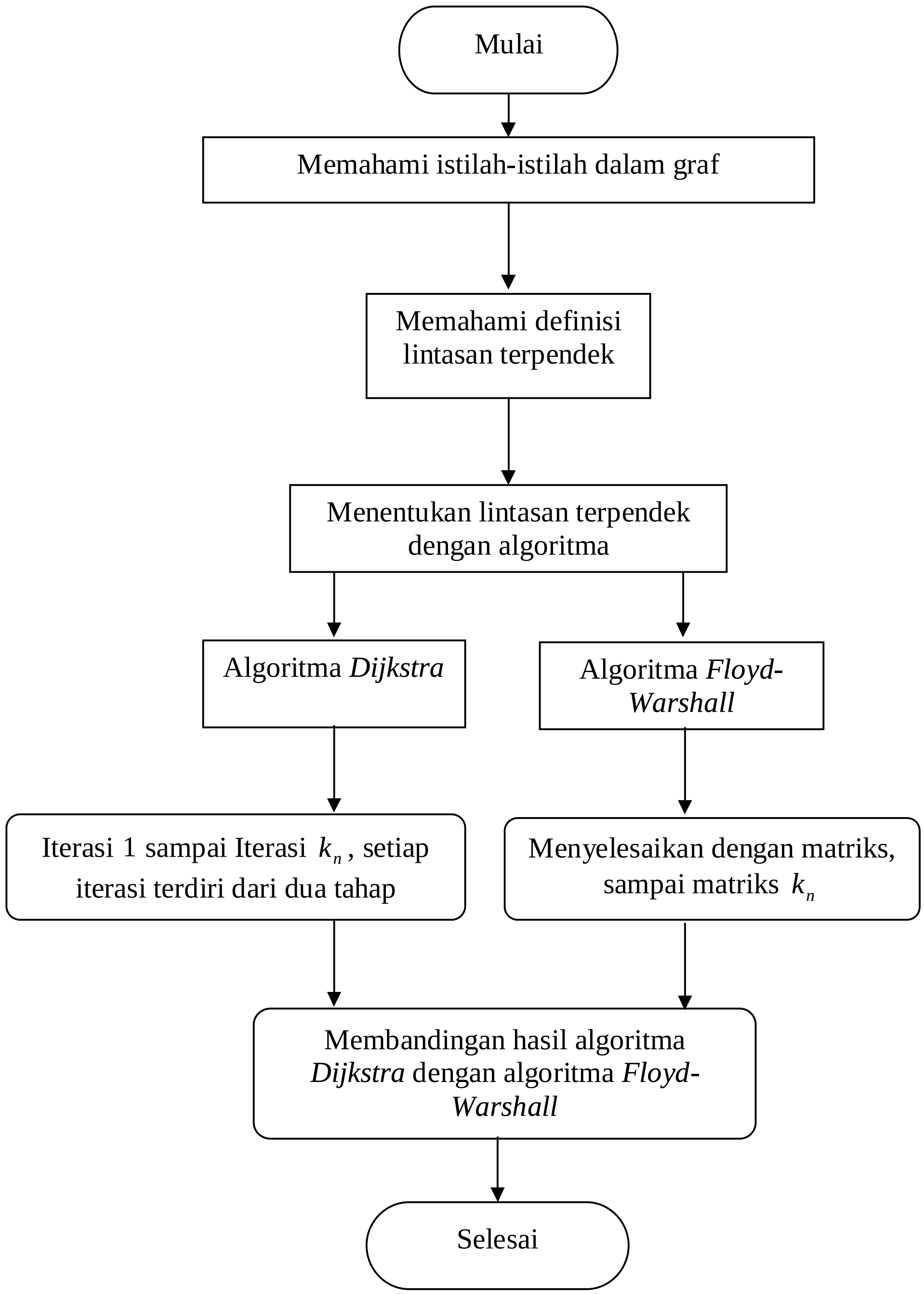
      Iterasi 1 sampai Iterasi k_n, setiap iterasi terdiri dari dua tahap

   b. algoritma *Floyd-Warshall* :

      Menyelesaikan dengan matriks, sampai matriks k_n

4. Membandingkan Algoritma *Dijkstra* dengan Algoritma *Floyd-Warshall*
5. Selesai.

Flowchart* perbandingan algoritma *Dijkstra* dan algoritma *Floyd-Warshall



Gambar 3.1 *Flowchart* Perbandingan antara algoritma *Dijkstra* dengan algoritma *Floyd-Warshall*

BAB IV

Bab ini berisikan penyelesaian masalah tugas akhir yang berjudul "**Membandingkan algoritma *Dijkstra* dan algoritma *Floyd-Warshall* untuk menentukan lintasan terpendek pada pendistribusian Koran Riau Pos"** adalah sebagai berikut :

Tabel 4.1 Jarak antar tempat pendistribusian koran Riau Pos

NO	Jalur Lintasan	Jarak Lurus (Meter)	Simbol Simpul
1	Jln. HR. Subrantas-Soekarno Hatta Atas (Riau Pos–Poto Kopi Falzafah Sp Kubang)	4500 M	$A-B$
2	Jln. Kaharudin Nst (Poto Kopi Falzafah Sp Kubang-Swalayan Planet Sp Pasir Putih)	2000 M	$B-C$
3	Jln. HR Subrantas–Adi Sucipto (Riau Pos - Poto Kopi Budi Sp 4 Rambutan)	2000 M	$A-F$
4	Jln. Kartama (Poto Kopi Budi Sp 4 Rambutan–Swalayan Planet Sp Pasir Putih)	5000 M	$F-C$
5	Jln. Adi Sucipto / Komp AURI (Poto Kopi Budi Sp Rambutan–Rumah Makan Lubuk Idai / Purna MTQ)	2500 M	$F-E$
6	Jln Jend. Sudirman (Rumah Makan Lubuk Idai/ Purna MTQ–Rumah Makan Pondok Patin Sp 3)	2000 M	$E-D$
7	Jln. Kaharudin Nst (Rumah Makan Pondok Patin Sp 3–Swalayan Planet Sp Pasir Putih)	1500 M	$D-C$
8	Jln. HR. Subrantas-Soekarno Hatta (Riau Pos – Citra Auto Sp Arifin Ahmad)	2500 M	$A-H$
9	Jln. Arifin Ahmad (Citra Auto Sp Soekarno Hatta-Kakandepag Pekanbaru-Sp Rambutun)	2500 M	$H-G$

10	Jln. Rambutan (Kakandepag Pekanbaru Sp Arifin Ahmad Rambutun - Poto Kopi Budi Sp 4 Rambutun)	3000 M	*G – F*

keterangan simpul :

A = Jln. HR. Subrantas-Soekarno Hatta Ujung (Riau Pos–Poto Kopi Falzafa Sp Kubang).

B = Jln. Kaharudin Nst (Poto Kopi Falzafa Sp Kubang-Swalayan Planet Sp Pasir Putih).

C = Swalayan Planet Sp Pasir Putih.

D = Jln. Kaharudin Nst (Rumah Makan Pondok Patin Sp 3–Swalayan Planet Sp Pasir Putih).

E = Jln Jend. Sudirman (Rumah Makan Lubuk Idai–Rumah Makan Pondok Patin Sp 3).

F = Jln. Adi Sucipto/ Komp AURI(Poto Kopi Budi Sp Rambutan–Rumah Makan Lubuk Idai / Purna MTQ).

G = Jln. Rambutan (Kakandepag Pekanbaru-Poto Kopi Budi Sp 4 Adi Sucipto).

H = Jln. HR. Subrantas-Soekarno Hatta (Riau Pos–Citra Auto Sp Arifin Ahmad).

4.1 Algoritma Dijkstra

Algoritma *Dijkstra* merupakan salah satu varian dari algoritma *Greedy*, yaitu salah satu bentuk algoritma populer dalam pemecahan persoalan yang terkait dengan masalah optimasi. Sifatnya sederhana dan gampang. Sesuai dengan artinya yang secara harafiah berarti tamak atau rakus namun tidak dalam konteks negatif (–), algoritma *Dijkstra* ini hanya memikirkan solusi terbaik yang akan diambil pada setiap langkah tanpa memikirkan konsekuensi ke depan.

Prinsipnya, ambillah apa yang bisa anda dapatkan saat ini (*take what you can get now!*), dan keputusan yang telah diambil pada setiap langkah tidak akan bisa diubah kembali. Intinya algoritma *Dijkstra* ini berupaya membuat pilihan nilai optimal mungkin pada setiap langkah.

4.2 Algoritma *Floyd-Warshall*

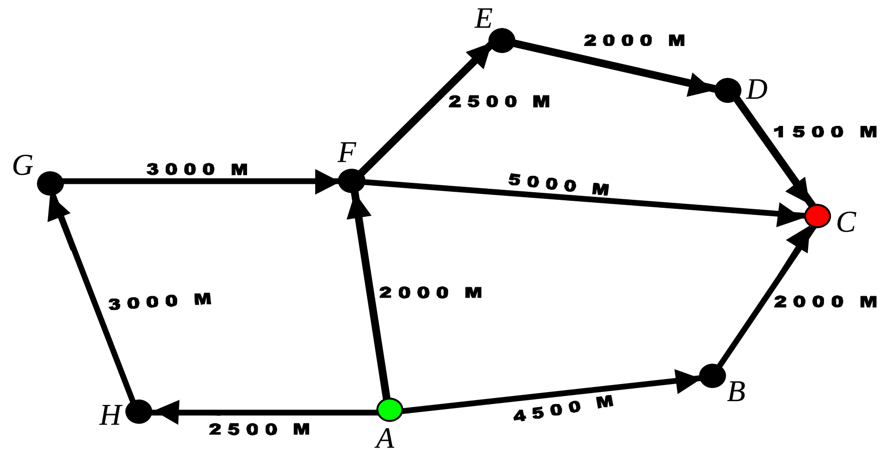
Algoritma *Floyd-Warshall* adalah salah satu varian dari pemrograman dinamis, yaitu suatu metode yang melakukan pemecahan masalah dengan memandang solusi yang akan diperoleh sebagai suatu keputusan yang saling terkait. Artinya solusi-solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya dan ada kemungkinan solusi lebih dari satu. Hal yang membedakan pencarian solusi menggunakan pemograman dinamis dengan algoritma *Dijkstra* adalah bahwa keputusan yang diambil pada tiap tahap pada algoritma *Dijkstra* hanya berdasarkan pada informasi yang terbatas sehingga nilai optimum yang diperoleh pada saat itu.

Jadi pada algoritma *Dijkstra* kita tidak memikirkan konsekuensi yang akan terjadi seandainya kita memilih suatu keputusan pada suatu tahap. dalam beberapa kasus, algoritma *Dijkstra* memberikan solusi terbaik yang dimilikinya. Disilah peran algoritma *Dijkstra* mencoba untuk memberikan solusi yang memiliki pemikiran terhadap konsekuensi yang ditimbulkan dari pengambilan keputusan pada suatu tahap. Algoritma *Dijkstra* mampu mengurangi pengeluaran keputusan yang tidak mengarah ke solusi. Prinsip yang dipegang oleh algoritma *Dijkstra* adalah prinsip optimalitas, yaitu jika solusi total optimal, maka bagian solusi sampai suatu tahap misalnya tahap ke-*i* juga optimal.

Algoritma *Floyd-Warshall* bekerja dengan menghitung *shortest Path* $(i, j, 1)$ untuk semua pasangan (i, j), kemudian hasil tersebut akan digunakan untuk menghitung *shortest Path* $(i, j, 2)$, untuk semua pasangan (i, j). Proses ini akan terus berlangsung hingga $k = n$ dan kita telah menemukan jalur terpendek untuk semua pasangan (i, j) menggunakan simpul-simpul perantara.

4.3 Penyelesaian Masalah

Keterhubungan antara kedelapan tempat pendistribusian Koran Riau Pos dapat dibentuk dan diselesaikan dengan menggunakan graf berbobot di bawah ini:



Gambar 4.1 Tempat pendistribusian koran Riau Pos

keterangan warna gambar 4.1 :

Hijau = Riau Pos

Merah = Swalayan Planet Sp Pasir Putih (tujuan utama pendistribusian)

Berdasarkan gambar keterhubungan antara kedelapan simpul (gambar 4.1) dipermasalahkan lintasan terpendek dari A ke C. Selanjutnya untuk penyelesaian masalah tersebut dapat disajikan ke dalam bentuk matriks (4.1) di bawah ini :

$$W = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array} \begin{array}{c} \begin{array}{cccccccc} A & B & C & D & E & F & G & H \end{array} \\ \begin{bmatrix} \infty & 45 & \infty & \infty & \infty & 20 & \infty & 25 \\ \infty & \infty & 20 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 15 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 20 & \infty & \infty & \infty & \infty \\ \infty & \infty & 50 & \infty & 25 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 30 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 30 & \infty \end{bmatrix} \end{array} \qquad (4.1)$$

Berdasarkan pada matriks (4.1), penyelesaian lintasan terpendek dari A ke C dapat diselesaikan dengan algoritma *Dijkstra* dan algoritma *Floyd-Warshall* sebagai berikut :

4.3.1 Penyelesaian dengan algoritma *Dijkstra*

Algoritma *Dijkstra* ini, pertama semua simpul yang termuat di dalam graf, yaitu A, B, C, D, E, F, G, H disusun ke dalam sebuah himpunan V sehingga terbentuklah sebuah himpunan $V = \{A, B, C, D, E, F, G, H\}$.

Selanjutnya dari himpunan V akan dipilih satu-persatu simpul, setelah terpilih, maka masuk ke dalam himpunan L. Proses pemilihan simpul di V berakhir, jika pada himpunan V menjadi himpunan kosong, sedangkan pada L menjadi $L = \{A, B, C, D, E, F, G, H\}$. Adapun proses pemilihan simpul di V langkah-langkahnya terurut seperti dibawah ini :

Iterasi 1.

Dipilih $A \in V$ sebagai simpul awal, maka diperoleh $L = \{A\}$ dan $V - \{A\} = \{B, C, D, E, F, G, H\}$. Selanjutnya simpul tersebut diproses melalui dua tahap sebagai berikut :

Tahap 1: Dipilih yang terendah dari :

$D(B) = W[A, B] = 45$; $D(F) = W[A, F] = 20$

$D(C) = W[A, C] = \infty$; $D(G) = W[A, G] = \infty$;

$D(D) = W[A, D] = \infty$; $D(H) = W[A, H] = 25$

$D(E) = W[A, E] = \infty$;

Diperoleh nilai terendah:

$D(F) = W[A, F] = 20$, sehingga

$L = \{A, F\}$ dan $V - L = \{B, C, D, E, G, H\}$.

Tahap 2: Diuji untuk setiap $j \in V - L$, apakah $D(j) > D(F) + W[F, j]$

untuk $j = B$,

$D(B) = W[A, B] = 45 < D(F) + W[F, B] = \infty$. Hal ini berarti pada matriks (4.1) tidak berubah $D(B) = 45$.

untuk $j = C$,

$D(C) = W[A, C] = \infty > D(F) + W[F, C] = 20 + 50 = 70$. Hal ini berarti pada matriks (4.1) terjadi perubahan $D(C) = 70$.

untuk $j = D$,

$D(D) = W[A, D] = \infty = D(F) + W[F, D] = \infty$. Hal ini berarti pada matriks (4.1) tidak berubah $D(D) = \infty$.

untuk $j = E$,

$D(E) = W[A, E] = \infty > D(F) + W[F, E] = 20 + 25 = 45$. Hal ini berarti pada matriks (4.1) terjadi perubahan $D(E) = 45$.

untuk $j = G$,

$D(G) = W[A, G] = \infty = D(F) + W[F, G] = 20 + \infty = \infty$. Hal ini berarti pada matriks (4.1) tidak berubah $D(G) = \infty$.

untuk $j = H$,

$D(H) = W[A, H] = 25 < D(F) + W[F, H] = \infty$. Hal ini berarti pada matriks (4.1) tidak berubah $D(H) = 25$.

Iterasi 2.

Berdasarkan dari iterasi 1, $F \in V$ terpilih, maka diperoleh $L = \{A, F\}$ sehingga terbentuk $V - L = \{B, C, D, E, G, H\}$. Selanjutnya simpul tersebut diproses melalui dua tahap sebagai berikut :

Tahap 1: Dipilih yang terendah dari :

$D(B) = W[A, B] = 45;\ D(E) = W[A, E] = 45;$

$D(C) = W[A, C] = 70;\ D(G) = W[A, G] = \infty;$

$D(D) = W[A, D] = \infty;\ D(H) = W[A, H] = 25$

Diperoleh nilai terendah $D(H) = 25$.

Tahap 2: Diuji untuk setiap $j \in V - L = \{B, C, D, E, G\}$. apakah

$D(j) > D(H) + W[H, j]$

untuk $j = B$,

$D(B) = 45 > D(H) + W[H, B] = 45 + \infty = \infty$. Hal ini berarti pada matriks (4.1) tidak berubah $D(B) = 45$.

untuk $j = C$,

$D(C) = 70 > D(H) + W[H, C] = 25 + \infty = \infty$. Hal ini berarti pada matriks (4.1) tidak berubah $D(C) = 70$.

untuk $j = D$,

$D(D) = D(H) + W[H, D] = \infty$ Hal ini berarti pada matriks (4.1) tidak berubah $D(D) = \infty$.

untuk $j = E$,

$D(E) = 45 > D(H) + W[H, E] = 45 + \infty = \infty$. Hal ini berarti pada matriks (4.1) tidak berubah $D(E) = 45$.

untuk $j = G$,

$D(G) = \infty > D(H) + W[H, G] = 25 + 30 = 55$. Hal ini berarti pada matriks (4.1) $D(G)$ berubah menjadi $D(G) = 55$.

Iterasi 3.

Berdasarkan dari iterasi 2, $H \in V$ terpilih, maka diperoleh $L = \{A, F, H\}$ sehingga terbentuk $V - L = \{B, C, D, E, G\}$. Selanjutnya simpul tersebut diproses melalui dua tahap sebagai berikut :

Tahap 1: Dipilih yang terendah dari :

$D(B) = [A, B] = 45$; $D(D) = [A, D] = \infty$; $D(G) = [A, G] = \infty$

$D(C) = [A, C] = 70$; $D(E) = [A, E] = 45$

Terendah $D(B) = [A, B] = 45$.

Tahap 2 :

Diuji untuk setiap $j \in V - L = \{C, D, E, G\}$, apakah $D(j) > D(B) + W[B, j]$.

untuk $j = C$,

$D(C) = 70 > D(B) + W[B,C] = 65$. Hal ini berarti pada matriks (4.1) terjadi perubahan $D(C) = 65$.

untuk $j = D,$

$D(D) = \infty = D(B) + W[H,D] = \infty$. Hal ini berarti pada matriks (4.1) tidak berubah $D(D) = \infty$.

untuk $j = E,$

$D(E) = 45 > D(B) + W[B,E] = \infty$. Hal ini berarti pada matriks (4.1) tidak berubah $D(E) = 45$.

untuk $j = G,$

$D(G) = 45 = D(B) + W[B,G] = \infty$. Hal ini berarti pada matriks (4.1) tidak berubah $D(G) = \infty$.

Iterasi 4.

Berdasarkan dari iterasi 3, $B \in V$ terpilih, maka diperoleh $L = \{A, F, H, B\}$ sehingga terbentuk $V - L = \{C, D, E, G\}$. Selanjutnya simpul tersebut diproses melalui dua tahap sebagai berikut :

Tahap 1 : Dipilih yang terendah dari :

$D(C) = 65;\ D(E) = 45;$

$D(D) = \infty;\ D(G) = \infty;$

Diperoleh nilai terendah $D(E) = [A,E] = 45$.

Tahap 2 :

Diuji untuk setiap $j \in V - L\{C, D, G\}$, apakah $D(j) > D(B) = W[B, j]$

untuk $j = C$,

$D(C) = 65 < D(E) + W[E,C] = 45 + \infty = \infty$. Hal ini berarti pada matriks (4.1) tidak terjadi perubahan $D(C) = 65$.

untuk $j = D,$

$D(D) = \infty > D(E) + W[E,D] = 45 + 20 = 65$. Hal ini berarti pada matriks (4.1) terjadi perubahan $D(D) = 65$.

untuk $j = G$,

$D(G) = \infty = D(D) + W[D, G] = \infty$. Hal ini berarti pada matriks (4.1) tidak berubah 1 $D(G) = \infty$.

Iterasi 5.

Berdasarkan dari iterasi 4, $G \in V$ terpilih, maka diperoleh $L = \{A, F, H, B, G\}$, sehingga terbentuk $V - L = \{C, D, E\}$. Selanjutnya simpul tersebut diproses melalui dua tahap sebagai berikut :

Tahap 1 : Dipilih yang terendah dari :

$D(C) = 65$; $D(D) = \infty$;

$D(E) = 45$;

Diperoleh nilai terendah :

$D(E) = 45$.

Tahap 2 :

Diuji untuk setiap $j \in V - L\{C, D,\}$, apakah $D(j) > D(E) = W[E, j]$.

untuk $j = C$,

$D(C) = 65 > D(C) + W[E, C] = 45 + \infty = \infty$. Hal ini berarti pada matriks (4.1) tidak berubah $D(C) = 65$.

untuk $j = D$,

$D(D) = \infty > D(E) + W[E, D] = 45 + 20 = 65$. Hal ini berarti pada matriks (4.1) terjadi perubahan $D(D) = 65$.

Iterasi 6.

Berdasarkan dari iterasi 5, $E \in V$ terpilih, maka diperoleh $L = \{A, F, H, B, G, E\}$, sehingga terbentuk $V - L = \{C, D\}$. Selanjutnya simpul tersebut diproses melalui dua tahap sebagai berikut :

Tahap 1 : Dipilih yang terendah dari :

$D(C) = 65$; $D(D) = 65$;

Diperoleh nilai terendah : $D(D) = 65$.

Tahap 2 :

Diuji $D(C) = 65 < D(D) + W[D, C] = 65 + 15 = 80$. Hal ini pada matriks (4.1) tidak berubah $D(C) = 65$.

Iterasi 7

Berdasarkan dari iterasi 7, $D \in V$ terpilih, maka diperoleh $L = \{A, F, H, B, G, E, D\}$, sehingga terbentuk $V - L = \{C\}$, maka terpilih $D(C) = 65$ terendah, sehingga $C \in V$. Karena $L = \{A, F, H, B, G, E, D, C\}$ dan $V = \{\ \}$, maka langka penyelusuran simpul selesai dan diperoleh $D(C) = 65$.

4.3.2 Penyelesaian dengan algoritma *Floyd-Warshall*

Algoritma *Floyd-Warshall* pemilihan simpul untuk mendapatkan nilai lintasan minimal dapat dilakukan secara bebas, artinya tidak terikat oleh simpul yang menghasilkan nilai lintasan yang lebih rendah, dalam uraian ini algoritma *Floyd-Warshall* diselesaikan langkah demi langkahnya sesuai dengan urutan kolom pada matriks dibawah ini :

$$W = W_0 \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array} \begin{array}{c} \begin{array}{cccccccc} A & B & C & D & E & F & G & H \end{array} \\ \begin{bmatrix} \infty & 45 & \infty & \infty & \infty & 20 & \infty & 25 \\ \infty & \infty & 20 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 15 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 20 & \infty & \infty & \infty & \infty \\ \infty & \infty & 50 & \infty & 25 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 30 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 30 & \infty \end{bmatrix} \end{array} \qquad (4.2)$$

Matriks tersebut membuat urutan kolom A, B, C, D, E, F, G dan H, ditulis sebagai himpunan $V = \{A, B, C, D, E, F, G, H\}$. Selanjutnya diperoleh langkah-langkah penyelesaian sebagai berikut :

a). Menentukan nilai pada kolom A.

Dicari penghubung $j \in V$ dengan A. Berdasarkan matriks (4.2) di atas, tidak terdapat penghubung $j \in V$ ke A, maka pada matriks (4.2) tidak terjadi perubahan $W[A, A] = \infty$.

b). Menentukan nilai pada kolom B.

Berdasarkan matriks (4.2) terdapat penghubung ke B,yaitu dari A ke B dengan $W[A, B] = 45$. Karena $W[A, B] < W[A, A] + W[A, B] = \infty + 45 = \infty$, maka pada matriks (4.2) tetap $W[A, B] = 45$. Karena tidak ada lagi penghubung pada matriks (4.2) ke B, maka matriks (4.2) tidak terjadi perubahan.

c). Menentukan nilai pada kolom C.

Berdasarkan matriks (4.2) terdapat penghubung ke C, yaitu dari B ke C dengan $W[B, C] = 20$ dapat dihitung $W[A, C] = W[A, B] + W[B, C] =$ 45+20 = 65. Karena $W[A, C] = \infty > 65$, maka pada matriks (4.2) berubah $W[A, C] = 65$. D ke C, dengan $W[D, C] = 15$ karena $W[D, B] + W[B, C] = \infty + 20 = \infty > W[D, C] = 15$, maka pada matriks (4.2) tetap $W[D, C] = 15$. F ke C dengan $W[F, C] = 50 < W[F, E] + W[E, C] = \infty$, maka $W[F, C] = 50$, tidak berubah. Selanjutnya diperoleh matriks (4.3) sebagai berikut :

$$
\begin{array}{c c}
 & \begin{array}{cccccccc} A & B & C & D & E & F & G & H \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array} &
\begin{bmatrix}
\infty & 45 & 65 & \infty & \infty & 20 & \infty & 25 \\
\infty & \infty & 20 & \infty & \infty & \infty & \infty & \infty \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
\infty & \infty & 15 & \infty & \infty & \infty & \infty & \infty \\
\infty & \infty & \infty & 20 & \infty & \infty & \infty & \infty \\
\infty & \infty & 50 & \infty & 25 & \infty & \infty & \infty \\
\infty & \infty & \infty & \infty & \infty & 30 & \infty & \infty \\
\infty & \infty & \infty & \infty & \infty & \infty & 30 & \infty
\end{bmatrix}
\end{array}
\qquad (4.3)
$$

d). Menentukan nilai pada kolom D.

Berdasarkan matriks (4.2) terdapat penghubung ke D, yaitu dari E ke D dengan $W[E,D]=20$, maka pada matriks (4.2) berubah $W[E,D]+W[D,C]=20+15=35<W[E,C]=\infty$, maka $W[E,C]=35$. Karena $W[F,E]+W[E,D]=25+20=45<W[F,D]=\infty$, maka pada matriks (4.2) berubah $W[F,D]=45$. Selanjutnya di peroleh matriks (4.4) sebagai berikut :

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array} \begin{array}{c} \begin{array}{cccccccc} A & B & C & D & E & F & G & H \end{array} \\ \begin{bmatrix} \infty & 45 & 65 & \infty & \infty & 20 & \infty & 25 \\ \infty & \infty & 20 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 15 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 35 & 20 & \infty & \infty & \infty & \infty \\ \infty & \infty & 50 & 45 & 25 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 30 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 30 & \infty \end{bmatrix} \end{array} \qquad (4.4)$$

e). Menentukan nilai pada kolom E.

Berdasarkan matriks (4.2) terdapat penghubung ke E, dari F ke E dengan $W[F,E]=25$. Karena $W[F,E]<W[F,D]+W[D,E]=45+\infty=\infty$, maka pada matriks (4.4) tetap $W[F,E]=25$. Karena $W[F,E]+W[E,C]=25+35=60<W[F,C]=50$, maka pada matriks (4.4) tidak berubah $W[E,C]=50$. Karena tidak ada lagi penghubung dari matriks (4.2) ke E, maka diperoleh matriks (4.5) sebagai berikut :

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array} \begin{array}{c} \begin{array}{cccccccc} A & B & C & D & E & F & G & H \end{array} \\ \begin{bmatrix} \infty & 45 & 65 & \infty & \infty & 20 & \infty & 25 \\ \infty & \infty & 20 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 15 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 35 & 20 & \infty & \infty & \infty & \infty \\ \infty & \infty & 60 & 45 & 25 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 30 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 30 & \infty \end{bmatrix} \end{array} \qquad (4.5)$$

f). Menentukan nilai pada kolom F .

Berdasarkan matriks (4.2) terdapat penghubung ke F , yaitu dari A ke F dengan $W[A,F]=20$, dan dapat dihitung, $W[A,F]+W[F,E]=20+25=45<$ $W[A,E]=\infty$, maka pada matriks (4.2) terjadi perubahan $W[A,E]=45$. Sebab $W[A,F]+W[F,D]=20+45=65<W[A,D]=\infty$, maka pada matriks (4.5) berubah $W[A,D]=65$. $W[A,F]+W[F,D]=20+45=65<W[A,D]=\infty$, maka pada matriks (4.5) berubah $W[A,D]=65$, dari G ke F dengan $W[G,F]=30$.

$W[G,F]+W[F,E]=30+25=55<W[G,E]=\infty$ maka pada matriks (4.5) berubah $W[G,E]=55$. $W[G,F]+W[F,D]=30+45=75<W[G,D]=\infty$, maka pada matriks (4.5) berubah $W[G,D]=75$. $W[G,F]+W[F,C]=30+50=80$ $<W[G,C]=\infty$, maka pada matriks (4.5) berubah $W[G,C]=80$. Karena $W[H,G]+W[G,F]=30+30=60=W[H,F]=\infty$, maka pada matriks (4.5) berubah $W[H,F]=60$. Selanjutnya diperoleh matriks (4.6) sebagai berikut:

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array}\begin{array}{c} \begin{array}{cccccccc} A & B & C & D & E & F & G & H \end{array} \\ \begin{bmatrix} \infty & 45 & 65 & 65 & 45 & 20 & \infty & 25 \\ \infty & \infty & 20 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 15 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 35 & 20 & \infty & \infty & \infty & \infty \\ \infty & \infty & 50 & 45 & 25 & \infty & \infty & \infty \\ \infty & \infty & 90 & 75 & 55 & 30 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 60 & 30 & \infty \end{bmatrix} \end{array} \quad (4.6)$$

g). Menentukan nilai pada kolom G .

Berdasarkan matriks (4.2) terdapat penghubung ke G , dari H ke G dengan $W[H,G]=30$. Karena $W[H,G]+W[G,E]=30+55=85<W[H,E]=\infty$, maka pada matriks (4.6) berubah $W[H,E]=85$. $W[H,G]+W[G,D]=$ $30+75=105<$ $W[H,D]=\infty$, maka pada matriks (4.6) berubah $W[H,D]=85$. Karena $W[H,G]+W[G,C]=30+90=120<W[H,C]=\infty$, maka pada matriks

(4.6) berubah $W[H,C]=120$. Karena tidak ada lagi penghubung pada matriks (4.2) ke G, maka diperoleh matriks (4.7) sebagai berikut :

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array} \begin{array}{c} \begin{array}{cccccccc} A & B & C & D & E & F & G & H \end{array} \\ \begin{bmatrix} \infty & 45 & 65 & 65 & 45 & 20 & \infty & 25 \\ \infty & \infty & 20 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 15 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 35 & 20 & \infty & \infty & \infty & \infty \\ \infty & \infty & 50 & 45 & 25 & \infty & \infty & \infty \\ \infty & \infty & 90 & 75 & 55 & 30 & \infty & \infty \\ \infty & \infty & 120 & 105 & 85 & 60 & 30 & \infty \end{bmatrix} \end{array} \tag{4.7}$$

h). Menentukan nilai pada kolom H.

Berdasarkan matriks (4.2) terdapat penghubung ke H, dari A ke H dengan $W[A,H]=25$. Karena $W[A,H]+W[H,G]=25+30=55<W[A,G]=\infty$, maka pada matriks (4.7) berubah $W[A,G]=55$. Karena tidak ada lagi penghubung pada matriks (4.2) ke H, maka proses pemilihan simpul di V berakhir dan diperoleh matriks (4.8) sebagai berikut :

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array} \begin{array}{c} \begin{array}{cccccccc} A & B & C & D & E & F & G & H \end{array} \\ \begin{bmatrix} \infty & 45 & 65 & 65 & 45 & 20 & 55 & 25 \\ \infty & \infty & 20 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 15 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 35 & 20 & \infty & \infty & \infty & \infty \\ \infty & \infty & 50 & 45 & 25 & \infty & \infty & \infty \\ \infty & \infty & 90 & 75 & 55 & 30 & \infty & \infty \\ \infty & \infty & 120 & 105 & 85 & 60 & 30 & \infty \end{bmatrix} \end{array} \tag{4.8}$$

Algoritma *Dijkstra* dan algoritma *Floyd-Warshall* adalah dua metode yang dapat digunakan untuk menentukan lintasan terpendek dari suatu simpul ke simpul lainnya di dalam sebuah graf berarah. Selanjutnya dari dua algoritma tersebut dapat ditentukan kesimpulan bahwa algoritma *Dijkstra* lebih efisien dibandingkan algoritma *Floyd-Warshall*. Berdasarkan matriks (4.8) di atas, maka diperoleh lintasan terpendek dari $A =$ Riau Pos ke swalayan planet di simpang pasir putih adalah lintasan dari riau pos menuju foto copi Falsafah simpang kubang dan dilanjutkan ke swalayan planet simpang pasir putih dengan jarak tempuh 6500 Meter.

BAB V
KESIMPULAN DAN SARAN

Berdasarkan dari uraian penggunaan algoritma *Dijkstra* dan algoritma *Floyd-Warshall* pada bab IV untuk menentukan lintasan terpendek pada pendistribusian Koran Riau Pos, maka diperoleh kesimpulan dan saran sebagai berikut :

5.1 Kesimpulan

Hasil lintasan terpendek pada permasalahan pendistribusian Koran Riau Pos dengan membandingkan algoritma *Dijkstra* dan algoritma *Floyd-Warshall* dari $A \to B \to C$ (Riau Pos ke Swalayan Planet di simpang pasir putih melewati poto kopi Falsafah di simpang kubang) adalah sama yaitu 4500+2000 = 6500 Meter.

Algoritma *Dijkstra* lebih efisien dikarenakan penelusuran simpul-simpul selalu mencari simpul yang berbobot minimum. Sehingga pada penelusuran tercapai simpul yang berbobot minimum, maka dapat diketahui lintasan dan bobot minimumnya. Sedangkan pada algoritma *Floyd-Warshall*, lintasan minimum dari simpul ke simpul yang dituju terlebih dahulu harus menyelesaikan tahap lintasan yang ada.

5.2 Saran

Sebagai saran yang ditujukan kepada pembaca yang ingin menentukan lintasan terpendek dengan menggunakan algoritma *Dijkstra* dan algoritma *Floyd-Warshall,* agar dapat mengembangkan atau menggunakan simpul-simpul yang lebih banyak lagi. Selanjutnya dibuat ke dalam bentuk program.

DAFTAR PUSTAKA

Erwin, K. *Matematika Teknik Lanjutan*. Edisi ke-6. Jakarta: PT.Gramedia Pustaka Utama. 1993.

https:// PAMA4208/MODUL 1/Pengantar. Teori.Graph.pdf .10 Maret 2011.

Munir, Rinaldi. *Matematika Diskrit*. Edisi ke-1. Bandung: PT. Informatika. 2001.

_____________. *Matematika Diskrit*. Edisi ke-2. Bandung: PT. Informatika. 2003.

Richard, Johnsonbaugh. *Matematika Diskrit*. jilid ke-2. Jakarta: PT. Prenhanllindo. 2002.

Siang, Jong Jek. *Matematika Diskrit dan Aplikas pada Ilmu Komputer*. Edisi ke-3. Yogyakarta: C.V ANDI OFFSET. 2006.

Suryadi, H.S. *Teori Graph Dasar*. Jakarta: Gunadarma. 1994.